1.

```
#include<stdio.h>
#include<stdlib.h>
void accept(int Arr[],int size)            //FOR ACCEPTING VALUE
{
        int i;
        printf("SIZE=%d",size);
        printf("\nENTER THE ELEMENT IN SET:\n");
        for(i=0;i<size;i++)
        {
                scanf("%d",&Arr[i]);
        }
}
void display(int Arr[],int size)                    //FOR DISPLAY VALUE
{
        int i;
        printf("\nTHE ELEMENT IN SET:\n");
        printf("{");
        for(i=0;i<size;i++)
        {
                if(i<size-1)
                {
                        printf("%d,",Arr[i]);
                }
                else if(i==size-1)
                {       printf("%d",Arr[i]);
                }
        }
        printf("}");
}
int uni(int A[],int m,int B[],int n,int UNO[])            //for union
{
        int i,j,l,k=0;
        for(l=0;l<m;l++)
        {
                UNO[k]=A[l];
                k++;
        }
        for(j=0;j<n;j++)
        {
                for(i=0;i<m;i++)
                {
                        if(B[j]==A[i])
                        break;
                }
                if(i==m)
                {
                        UNO[k]=B[j];
                        k++;
                }
```

```
	}
	return(k);
}
int intersection(int A[],int m,int B[],int n,int INT[])		//for intersection
{
	int i,j,l,k=0;
	for(j=0;j<n;j++)
	{
		for(i=0;i<m;i++)
		{
			if(B[j]==A[i])
			{	break;
			}
		}
		if(i!=m)
		{
			INT[k++]=B[j];
		}
	}
	return(k);
}
int difference(int A[],int m,int B[],int n,int DIF[])		//for difference
{
	int i,j,k=0;
	for(j=0;j<n;j++)
	{
		for(i=0;i<m;i++)
		{
			if(B[j]==A[i])
			{
				break;
			}
		}
		if(i==m)
		{
			DIF[k++]=B[j];
		}
	}
	return(k);
}
int sdiff(int DIFBA[],int difba,int DIFAB[],int difab,int SDIF[])	//for symmetric difference
{
	int k;
	k=uni(DIFAB,difab,DIFBA,difba,SDIF);
	return(k);
}
void show(int A[],int m,int B[],int n)		//for showing output
{
	printf("\nSET A:");
	display(A,m);
```

```
	printf("\nSET B:");
	display(B,n);
}
int main()
{
	int A[20],B[20],UNO[50],DIFAB[50],DIFBA[50],INT[50],SDIF[50];
	int m,n,uno,difab,difba,in,sd;
	system("clear");
	printf("\nENTER THE NUMBER OF ELEMENT IN SET A: ");
	scanf("%d",&m);
	accept(A,m);
	display(A,m);

	printf("\nENTER THE NUMBER OF ELEMENT IN SET B: ");
	scanf("%d",&n);
	accept(B,n);
	display(B,n);

	getchar();
	system("clear");

	//union
	show(A,m,B,n);
	uno=uni(A,m,B,n,UNO);
	printf("\nAFTER UNION:");
	display(UNO,uno);

	//intersection
	in=intersection(A,m,B,n,INT);
	printf("\nAFTER INTERSECTION:");
	display(INT,in);

	//difference
	difba=difference(B,n,A,m,DIFBA);
	printf("\nAFTER DIFFERENCE A-B:");
	display(DIFBA,difba);
	difab=difference(A,m,B,n,DIFAB);
	printf("\nAFTER DIFFERENCE B-A:");
	display(DIFAB,difab);

	//symetric differece
	sd=sdiff(DIFBA,difba,DIFAB,difab,SDIF);
	printf("\nAFTER SYMETRIC DIFFERENCE:");
	display(SDIF,sd);

	return 0;
}
```

Output:

```
C:\Users\DiL MGr\Documents\Desktop 2\Discrete Structure\LAB\LAB 1 solution\Lab 1(Q.1).exe

ENTER THE NUMBER OF ELEMENT IN SET A: 5
SIZE=5
ENTER THE ELEMENT IN SET:
1
2
3
4
5

THE ELEMENT IN SET:
{1,2,3,4,5}
ENTER THE NUMBER OF ELEMENT IN SET B: 5
SIZE=5
ENTER THE ELEMENT IN SET:
4
5
6
7
8
```

```
C:\Users\DiL MGr\Documents\Desktop 2\Discrete Structure\LAB\LAB 1 solution\Lab 1(Q.1).exe

SET A:
THE ELEMENT IN SET:
{1,2,3,4,5}
SET B:
THE ELEMENT IN SET:
{4,5,6,7,8}
AFTER UNION:
THE ELEMENT IN SET:
{1,2,3,4,5,6,7,8}
AFTER INTERSECTION:
THE ELEMENT IN SET:
{4,5}
AFTER DIFFERENCE A-B:
THE ELEMENT IN SET:
{1,2,3}
AFTER DIFFERENCE B-A:
THE ELEMENT IN SET:
{6,7,8}
AFTER SYMETRIC DIFFERENCE:
THE ELEMENT IN SET:
{6,7,8,1,2,3}
--------------------------------
Process exited after 21.85 seconds with return value 0
Press any key to continue . . .
```

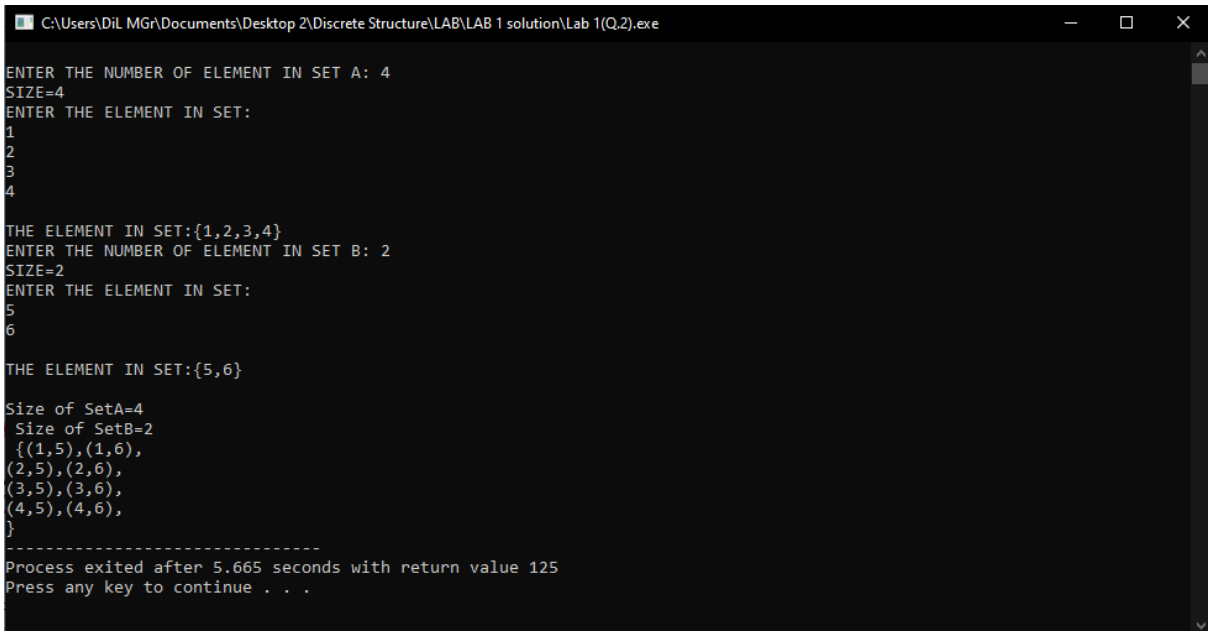
2.

```
#include<stdio.h>

void Cart(int [], int [], int, int);
void accept(int [], int);
void display(int [], int);
void bubbleSort(int [],int);
int main()
{
        int p,q;
        int a[20];
        int b[20];
```

```
	printf("\nENTER THE NUMBER OF ELEMENT IN SET A: ");
	scanf("%d",&p);
	accept(a,p);
	display(a,p);

	printf("\nENTER THE NUMBER OF ELEMENT IN SET B: ");
	scanf("%d",&q);
	accept(b,q);
	display(b,q);

	printf("\n\nSize of SetA=%d \n Size of SetB=%d",p,q);
	bubbleSort(a,p);
	bubbleSort(b,q);
	printf("\n ");
	Cart(a,b,p,q);
}

void accept(int Arr[],int size)		//FOR ACCEPTING VALUE
{
	int i;
	printf("SIZE=%d",size);
	printf("\nENTER THE ELEMENT IN SET:\n");
	for(i=0;i<size;i++)
	{
		scanf("%d",&Arr[i]);
	}
}

void display(int Arr[],int size)		//FOR DISPLAY VALUE
{
	int i;
	printf("\nTHE ELEMENT IN SET:");
	printf("{");
	for(i=0;i<size;i++)
	{
		if(i<size-1)
		{
			printf("%d,",Arr[i]);
		}
		else if(i==size-1)
		{	printf("%d}",Arr[i]);
		}
	}
}

void Cart(int a[],int b[],int m, int n)
{
   int i,j;
   printf("{");
   for(i=0;i<m;i++)
```

```
    {
        for(j=0;j<n;j++)
        {
            printf("(%d,%d)",a[i],b[j]);
            printf(",");
        }
        printf("\n");
    }
    printf("}");
}
void bubbleSort(int arr[], int n)
{
   int i, j,temp;
   for (i = 0; i < n-1; i++)
    {
      for (j = 0; j < n-i-1; j++)
      {

         if (arr[j] > arr[j+1])
         {
            temp=arr[j];
            arr[j]=arr[j+1];
            arr[j+1]=temp;
         }
      }
    }
}
```

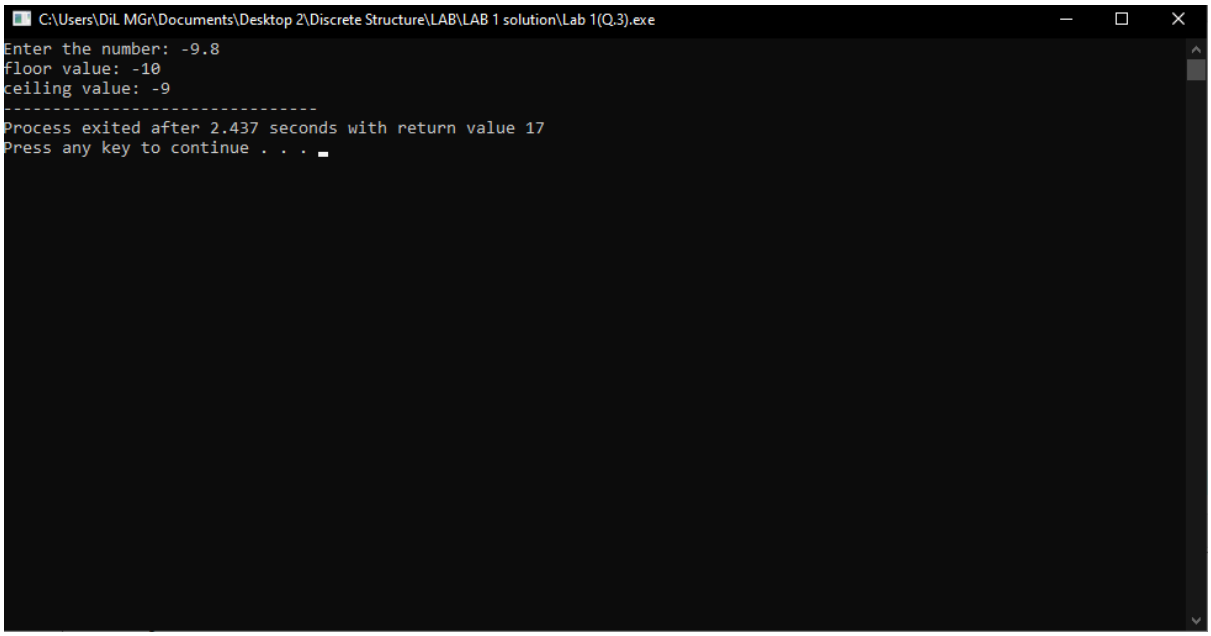
Output:



3.

```
#include<stdio.h>
void floor_ceiling(float num)
{
	if(num == (int)num)
	{
		printf("floor value: %d\n",(int)num);
		printf("ceiling value: %d",(int)num);
	}
	else
	{
		if(num >= 0)
		{
			printf("floor value: %d\n",(int)num);
			printf("ceiling value: %d",(int)(num+1));
		}
		else
		{
			printf("floor value: %d\n",(int)(num-1));
			printf("ceiling value: %d",(int)num);
		}
	}
}
int main()
{
	float num;
	printf("Enter the number: ");
	scanf("%f",&num);
	floor_ceiling(num);
}
```

Output:



4.

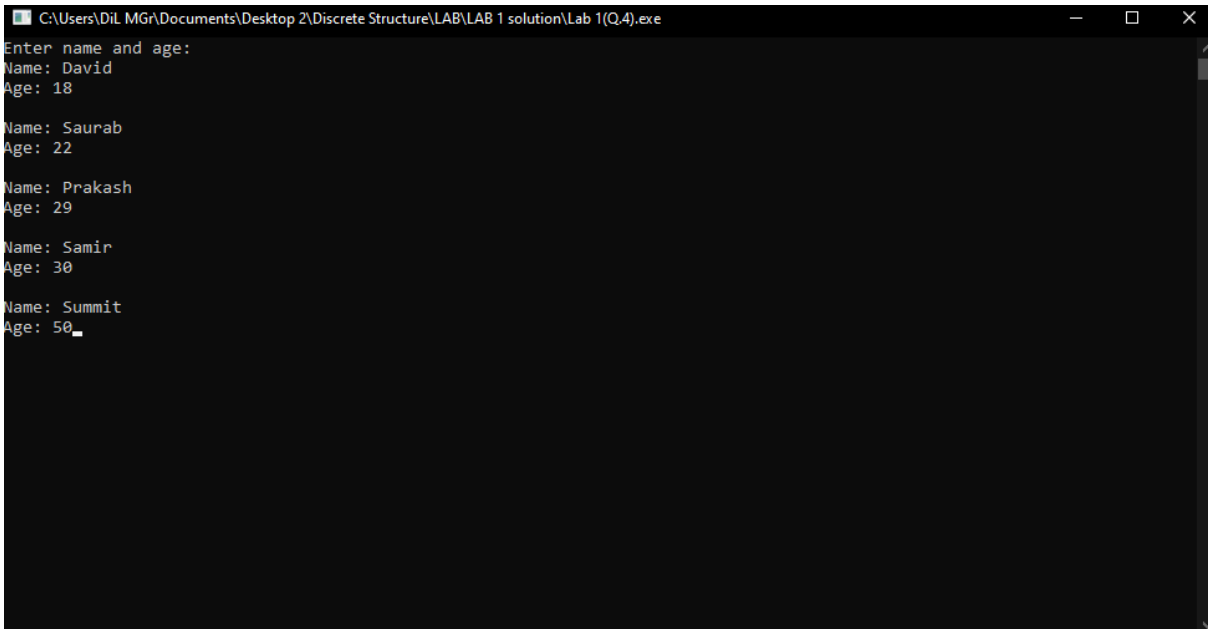
```
#include<stdio.h>
#include<stdlib.h>
void degree_of_membershipA(float [], int []);
void degree_of_membershipB(float [], int []);
int main()
{
	char name[40][40];
	int i,j,age[10];
	float degOfMemA[10],degOfMemB[10];

	printf("Enter name and age:\n");
	for(i=0;i<5;i++)
	{
		printf("Name: ");
		scanf("%s",&name[i]);
		printf("Age: ");
		scanf("%d",&age[i]);
		printf("\n");
	}
	system("clear");
	printf("Following are the name and age of 5 people:\n");
	for(i=0;i<5;i++)
	{
		printf("Name: %s and Age: %d\n",name[i],age[i]);
	}
	degree_of_membershipA(degOfMemA,age);
	degree_of_membershipB(degOfMemB,age);

	display(degOfMemA,"First Fuzzy Sets");
	display(degOfMemB,"Second Fuzzy Sets");

	fuzzy_union(name,degOfMemA,degOfMemB);
	fuzzy_intersection(name,degOfMemA,degOfMemB);
	fuzzy_complement(name,degOfMemA,"First Fuzzy sets");
	fuzzy_complement(name,degOfMemB,"Second Fuzzy sets");
}

void display(float degOfMem[10],char set[30])
{
	int i;
	printf("\nDegree of Membeship of %s = {",set);
	for(i=0;i<5;i++)
	{
		if(i<4)
		{
			printf("%.1f, ",degOfMem[i]);
		}
		else
		{
```

```
                printf("%.1f",degOfMem[i]);
            }
        }
        printf("}\n");
}
void degree_of_membershipA(float degOfMemA[10], int age[10])
{
        int i;
        float num;
        for(i=0;i<5;i++)
        {
            if(age[i]<=20)
            {
                degOfMemA[i] = 1;
            }
            else if(age[i] <= 30)
            {
                num = 30-age[i];
                degOfMemA[i] = num/10;
            }
            else
            {
                degOfMemA[i] = 0;
            }

        }
}
void degree_of_membershipB(float degOfMemB[10], int Age[10])
{
        int i;
        float num;
        for(i=0;i<5;i++)
        {
            if(Age[i]<=15)
            {
                degOfMemB[i] = 1;
            }
            else if(Age[i]>15 && Age[i]<=35)
            {
                num = 35-Age[i];
                degOfMemB[i] = num/20;
            }
            else
            {
                degOfMemB[i] = 0;
            }

        }
}
void fuzzy_union(char name[40][40],float degOfMemA[10],float degOfMemB[10])
```

```
{
	int i;
	float fuzzyUni[10];
	for(i=0;i<5;i++)
	{
		if(degOfMemA[i] > degOfMemB[i])
		{
			fuzzyUni[i] = degOfMemA[i];
		}
		else
		{
			fuzzyUni[i] = degOfMemB[i];
		}
	}
	printf("\nFuzzy Union = {");
	for(i=0;i<5;i++)
	{
		if(i<4)
		{
			printf("%.1f/%s, ",fuzzyUni[i],name[i]);
		}
		else
		{
			printf("%.1f/%s",fuzzyUni[i],name[i]);
		}
	}
	printf("}\n");
}
void fuzzy_intersection(char name[40][40],float degOfMemA[10],float degOfMemB[10])
{
	int i;
	float fuzzyIni[10];
	for(i=0;i<5;i++)
	{
		if(degOfMemA[i] > degOfMemB[i])
		{
			fuzzyIni[i] = degOfMemB[i];
		}
		else
		{
			fuzzyIni[i] = degOfMemA[i];
		}
	}
	printf("\nFuzzy Intersection = {");
	for(i=0;i<5;i++)
	{
		if(i<4)
		{
			printf("%.1f/%s, ",fuzzyIni[i],name[i]);
		}
```

```
		else
		{
			printf("%.1f/%s",fuzzyIni[i],name[i]);
		}
	}
	printf("}\n");
}
void fuzzy_complement(char name[40][40],float degOfMem[10],char set[20])
{
	int i;
	float fuzzyComp[10];
	for(i=0;i<5;i++)
	{
		fuzzyComp[i] = 1-degOfMem[i];
	}
	printf("\nFuzzy Complement of %s= {",set);
	for(i=0;i<5;i++)
	{
		if(i<4)
		{
			printf("%.1f/%s, ",fuzzyComp[i],name[i]);
		}
		else
		{
			printf("%.1f/%s",fuzzyComp[i],name[i]);
		}
	}
	printf("}\n");
}
```

Output:



```
C:\Users\DiL MGr\Documents\Desktop 2\Discrete Structure\LAB\LAB 1 solution\Lab 1(Q.4).exe
Following are the name and age of 5 people:
Name: David and Age: 18
Name: Saurab and Age: 22
Name: Prakash and Age: 29
Name: Samir and Age: 30
Name: Summit and Age: 50

Degree of Membeship of First Fuzzy Sets = {1.0, 0.8, 0.1, 0.0, 0.0}

Degree of Membeship of Second Fuzzy Sets = {0.9, 0.6, 0.3, 0.3, 0.0}

Fuzzy Union = {1.0/David, 0.8/Saurab, 0.3/Prakash, 0.3/Samir, 0.0/Summit}

Fuzzy Intersection = {0.9/David, 0.6/Saurab, 0.1/Prakash, 0.0/Samir, 0.0/Summit}

Fuzzy Complement of First Fuzzy sets= {0.0/David, 0.2/Saurab, 0.9/Prakash, 1.0/Samir, 1.0/Summit}

Fuzzy Complement of Second Fuzzy sets= {0.1/David, 0.4/Saurab, 0.7/Prakash, 0.8/Samir, 1.0/Summit}

--------------------------------
Process exited after 24.25 seconds with return value 0
Press any key to continue . . .
```